Capitolo 1

Strumenti per iniziare

di Michael Morrison

IN QUESTO CAPITOLO

- ✓ Il Java Development Kit 1.1 4
- ✓ Scelta di un IDE 12
- ✓ Symantec Café 14
- ✓ SunSoft Java WorkShop 15
- ✓ Microsoft Visual J++ 17
- ✓ Asymetrix SuperCede 17
- ✓ Roaster 18
- ✓ SourceCraft NetCraft 19
- ✓ Pro-C WinGEN per Java 19
- ✓ Rogue Wave JFactory 20
- ✓ Metrowerks CodeWarrior 21
- ✓ Riepilogo 22

Quando il linguaggio Java è stato presentato, nel 1995, l'unico strumento di sviluppo disponibile era il Java Development Kit (JDK) della Sun. Questa serie di strumenti a riga di comando consente di scrivere, compilare e verificare i programmi Java, ma è lontano da ambienti di sviluppo integrati come Visual Basic e Borland C++. Un *ambiente di sviluppo integrato* (IDE, Integrated Development Environment) è un software che raccoglie diversi strumenti di sviluppo in un unico pacchetto che solitamente include un editor, un compilatore, un debugger e altri strumenti utili per lo sviluppo. Nella maggior parte dei casi si tratta di pacchetti basati sulla grafica e le finestre, e sull'impiego del mouse: lo scopo è quello di rendere lo sviluppo di software più veloce, efficiente e facile da verificare.

Molti ambienti di sviluppo integrati utilizzano metodi di tipo RAD (Rapid Application Development), progettati per velocizzare il processo di sviluppo tramite l'uso di prototipi e strumenti per il disegno dell'interfaccia. Per gran parte degli ambienti di sviluppo di Java, gli

strumenti RAD sono dedicati alla costruzione dell'interfaccia utente grafica, e alcuni offrono una connessione diretta con il codice sorgente in modo che sia possibile disegnare un componente, ad esempio un pulsante, e passare direttamente al codice di gestione di eventi per associargli un'azione.

In questo capitolo sono trattati gli strumenti di sviluppo del JDK 1.1 standard e alcuni dei più noti ambienti di sviluppo integrato, elencati di seguito.

- ✔ Java Development Kit 1.1.
- ✔ Symantec Café.
- ✔ SunSoft Java WorkShop.
- ✔ Microsoft Visual J++.
- ✔ Asymetrix SuperCede.
- ✔ Roaster Technologies'Roaster.
- ✔ SourceCraft NetCraft.
- ✔ Pro-C WinGEN per Java.
- ✔ Rogue Wave JFactory.
- ✔ Metrowerks Code Warrior.

Il Java Development Kit 1.1

L'ultima versione del JDK è sempre disponibile presso il sito Web della Sun, `http://www.javasoft.com/`. Si consiglia di consultare questo sito anche per la documentazione che mette a disposizione.

Il JDK contiene una varietà di strumenti e informazioni su Java, tra cui i seguenti.

- ✔ Interprete di runtime.
- ✔ Compilatore.
- ✔ Visualizzatore di applet.
- ✔ Debugger.
- ✔ Disassemblatore di file di classe.
- ✔ Generatore di file di intestazione e di modello.
- ✔ Archiviatore.
- ✔ Gestore di firme digitali.
- ✔ Strumenti per la chiamata remota di metodi.

- ✔ Esempi e codice sorgente.
- ✔ Codice sorgente dell'API.

L'interprete di runtime è il cuore del sistema di Java; il compilatore, il visualizzatore di applet, il debugger, il disassemblatore di file di classe, il generatore di file di intestazione e di modello e il generatore di documentazione sono gli strumenti principali utilizzati dagli sviluppatori. Gli esempi mostrano interessanti applet Java, completi di codice sorgente. Infine, per chi è interessato ad approfondire la conoscenza di Java, il codice sorgente dell'API è una risorsa insostituibile.

L'interprete di runtime

L'interprete di runtime (`java`) è utilizzato per eseguire programmi eseguibili indipendenti in formato bytecode, compilato. Si tratta di uno strumento a riga di comando per eseguire *applicazioni*, ovvero programmi Java non grafici o con una finestra propria, mentre gli *applet* sono programmi grafici che richiedono il supporto di un browser Web e sono eseguiti interamente in esso. Ecco la sintassi per l'utilizzo dell'interprete di runtime:

```
java Opzioni NomeClasse Argomenti
```

L'argomento *`NomeClasse`* specifica il nome della classe che si desidera eseguire; se la classe si trova in un package, occorre specificare il nome completo. Per ulteriori informazioni si rimanda al Capitolo 4. Ad esempio, per eseguire una classe denominata *`Asteroidi`* che si trova in un package denominato *`GiochiAzione`*, si digita quanto segue:

```
java GiochiAzione.Asteroidi
```

Quando l'interprete Java esegue una classe, in realtà esegue il metodo `main()` di quest'ultima, e termina quando tale metodo e i thread creati da questo sono terminati. Non è il caso di approfondire ora questo argomento, che sarà trattato più avanti. Il metodo `main()` accetta un elenco di argomenti (*`Argomenti`*) utilizzabili per controllare il programma, che vanno passati all'interprete nella riga di comando.

Ad esempio, se si ha una classe denominata `FiltroTesto` che esegue un filtro su un file di testo, occorre specificare il nome di quest'ultimo:

```
java FiltroTesto UnFile.txt
```

L'argomento *`Opzioni`* specifica le opzioni che determinano il modo in cui l'interprete esegue il programma. Per ulteriori informazioni si consulti la documentazione.

Il compilatore

Il compilatore (`javac`) è utilizzato per compilare file di codice sorgente di Java in classi di bytecode eseguibili. I file di codice sorgente hanno estensione `.java`, mentre quelli generati dal compilatore hanno estensione `.class`. Il compilatore crea un unico file per ogni classe definita nel file sorgente.

Alcuni sistemi operativi non sono in grado di gestire le estensioni di Java perché sono troppo lunghe. Ad esempio, Windows 3.1 non permette di utilizzare estensioni lunghe più di tre caratteri, perciò occorre impiegare `.jav` e `.class`.

Il compilatore funziona a riga di comando, in modo simile all'interprete, con la seguente sintassi:

```
javac Opzioni NomeFile
```

L'argomento `NomeFile` specifica il nome del file sorgente da compilare, `Opzioni` le opzioni relative al modo in cui il compilatore crea le classi Java eseguibili. Si rimanda alla documentazione di Java per ulteriori informazioni

Il visualizzatore di applet

Il visualizzatore di applet (`appletviewer`) serve per controllare l'esecuzione degli applet di Java, invece di affidarsi a un browser Web. Ecco come si utilizza:

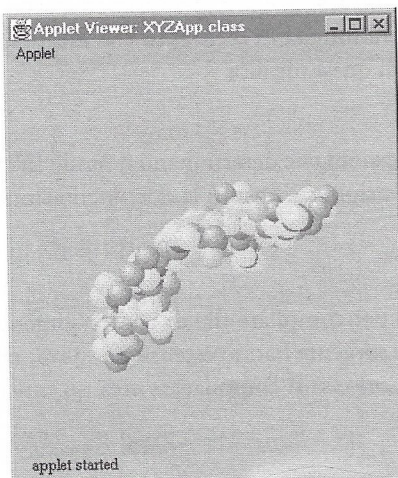
```
appletviewer Opzioni URL
```

L'argomento `URL` specifica l'indirizzo (URL) di un documento contenente una pagina HTML con un applet Java incorporato. `Opzioni` specifica il modo in cui eseguire l'applet. In realtà l'unica opzione disponibile è `-debug`, che avvia il debugger di Java. Nella Figura 1.1 è illustrato il visualizzatore di applet in azione con un esempio fornito nel JDK (`MoleculeViewer`).

Il debugger

Il debugger di Java (`jdb`) è un programma a riga di comando che consente di effettuare il debugging delle applicazioni Java, sfruttando l'API Java Debugger per fornire il supporto al debugging all'interno dell'interprete. Ecco la sintassi:

```
jdb Opzioni
```

Figura 1.1 *L'applet MoleculeViewer eseguito nell'Appletviewer di Java.*

L'argomento *Opzioni* è utilizzato per specificare diverse impostazioni in una sessione di debugging. Per ulteriori informazioni si rimanda al Capitolo 27.

Il disassemblatore di file di classe

Il disassemblatore di file di classe (javap) è utilizzato per disassemblare gli eseguibili. L'output di default è costituito da dati e metodi pubblici della classe; è uno strumento utile quando non si dispone del codice sorgente di una classe, ma si desiderano informazioni sull'implementazione. Ecco come si utilizza:

```
javap Opzioni NomiClassi
```

L'argomento *NomiClassi* specifica i nomi di una o più classi da disassemblare, mentre *Opzioni* indica il modo in cui vanno disassemblate.

Il generatore di file di intestazione e di modello

Il generatore di file di intestazione e di modello (javah) è utilizzato per generare file di intestazione e file sorgente per l'implementazione di metodi Java in C. I file generati possono essere utilizzati per accedere a variabili membro di un oggetto dal codice C. Il programma opera generando una struttura C corrispondente alla classe Java. Ecco come si utilizza:

```
javah Opzioni NomeClasse
```

L'argomento *NomeClasse* è il nome della classe da cui generare i file sorgente in C, *Opzioni* specifica in che modo generare i file.

Il generatore di documentazione

Il generatore di documentazione di Java (javadoc) è utile per generare documentazione direttamente dal codice sorgente. Esso analizza i file sorgente e genera pagine HTML in base a dichiarazioni e commenti. Ecco come si utilizza:

```
javadoc Opzioni NomeFile
```

L'argomento *NomeFile* specifica un package o un file sorgente di Java; nel primo caso, viene generata la documentazione di tutte le classi contenute nel package. L'argomento *Opzioni* determina il modo in cui viene eseguito javadoc.

L'archiviatore

L'archiviatore di Java (jar) è uno strumento utilizzato per raccogliere e comprimere più file (solitamente applet o applicazioni) in un singolo file di archivio, detto anche *file JAR*. Riunendo i componenti di un applet o un'applicazione in un singolo archivio, si facilita il loro trasferimento da parte del browser in un'unica transazione HTTP, invece di creare una nuova connessione per ogni file. Ciò, insieme alla compressione, riduce notevolmente i tempi di trasferimento. In più, l'archiviatore può essere utilizzato con il programma per la firma digitale (descritto più avanti) per firmare applet e applicazioni in modo che possano essere autenticate al momento dell'esecuzione.

Ecco la sintassi:

```
jar Opzioni NomeFileManifest NomeFileOutput NomiFileInput
```

L'argomento *NomeFileManifest* specifica un file manifest utilizzato per descrivere il contenuto dell'archivio creato. *NomeFileOutput* indica il nome dell'archivio da creare, *NomiFileInput* i file da aggiungere all'archivio. L'argomento *Opzioni* specifica la modalità di esecuzione del programma. Questi argomenti sono trattati in dettaglio nella Parte 8.

Il programma per la firma digitale

Il programma per la firma digitale (`javakey`), noto anche come lo strumento di sicurezza di Java, genera firme digitali per file di archivio. Le firme sono utilizzate per verificare che un file proviene da un'entità specifica, il *firmatario*. Per generare una firma, occorre prima associare il firmatario con una coppia di chiavi pubblica e privata e autenticare la chiave pubblica con un certificato. Il firmatario ha la responsabilità di gestire tutto ciò. Ecco come si utilizza il programma:

```
javakey Opzioni
```

L'argomento *Opzioni* controlla il funzionamento di javakey. Altre informazioni sulle firme digitali sono riportate nel Capitolo 32.

Strumenti per la chiamata remota di metodi

Il JDK comprende tre diversi strumenti per lavorare con RMI (Remote Method Invocation), costituiti da un compilatore di un file di modello, un registro di oggetti remoti e uno strumento di versione seriale. Per ulteriori informazioni si rimanda al Capitolo 16.

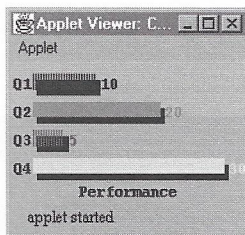
Esempi e codice sorgente

Il JDK è fornito con numerosi esempi, completi di codice sorgente.

- ✔ Animator
- ✔ ArcTest
- ✔ BarChart
- ✔ Blink
- ✔ BouncingHeads
- ✔ CardTest
- ✔ DitherTest
- ✔ DrawTest
- ✔ Fractal
- ✔ GraphicsTest

- ✔ GraphLayout
- ✔ ImageMap
- ✔ ImageTest
- ✔ JumpingBox
- ✔ MoleculeViewer
- ✔ NervousText
- ✔ ScrollingImages
- ✔ SimpleGraph
- ✔ SpreadSheet
- ✔ TicTacToe
- ✔ TumblingDuke
- ✔ UnderConstruction
- ✔ WireFrame

Non è il caso di descrivere tutti questi esempi, ma è utile mostrarne alcuni. Il primo applet interessante è `BarChart`, illustrato nella Figura 1.2. Si tratta di un buon esempio del modo in cui si può utilizzare Java per mostrare informazioni statistiche graficamente sul Web. I dati rappresentati dal grafico a barre potrebbero essere collegati a una fonte di dati in tempo reale, come delle quotazioni azionarie; in questo modo si potrebbe generare una rappresentazione grafica in tempo reale del proprio portafoglio di azioni.

Figura 1.2
L'applet BarChart.

L'applet `GraphicTest` (Figura 1.3) è un buon esempio di utilizzo della grafica. Java comprende un esteso set di funzioni grafiche, tra cui il supporto per forme primitive e routine di disegno più elaborate.

Rimanendo nel campo della grafica, l'applet `SimpleGraph` mostra come tracciare un diagramma bidimensionale, utile in campo scientifico ed educativo (Figura 1.4).

Nel campo commerciale, nulla è più utile di un buon foglio elettronico; l'applet `SpreadSheet` mostra come implementarne uno in Java (Figura 1.5).

Figura 1.3
L'applet GraphicsTest.

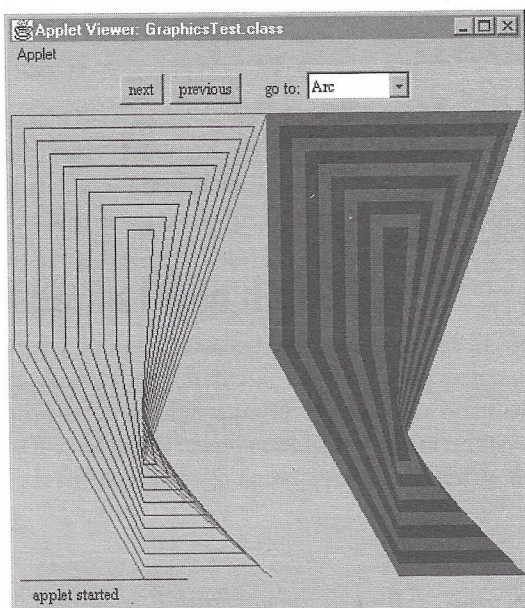



Figura 1.4
L'applet SimpleGraph.

Applet Viewer: SpreadSheet.class
Applet
Example

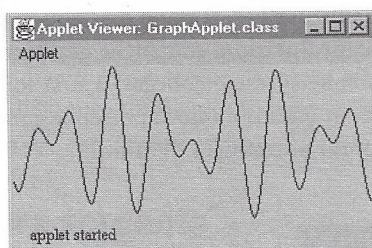
	A	B	C	D	E
1	10	500	10000		
2	30	1000	30000		
3			40000		
4					

applet started

Figura 1.5
L'applet SpreadSheet.

Per divertirsi un po' si può utilizzare l'applet `TicTacToe`, per giocare a tris con Java (Figura 1.6). L'ultimo applet che è il caso di citare qui è `UnderConstruction`, utile per le pagine Web ancora in fase di realizzazione: visualizza la mascotte di Java, Duke, e riproduce dei suoni (Figura 1.7).

L'interesse principale di questi applet di esempio è dato dalla disponibilità del codice sorgente, che consente di comprenderne il funzionamento per poter utilizzare tecniche simili nei propri programmi Java. Il modo migliore di apprendere è proprio quello di partire da esempi concreti.

Codice sorgente dell'API

L'ultimo componente del JDK è il codice sorgente dell'API, per tutte le classi che la costituiscono. La Sun non si è preoccupata di mantenere il segreto sui dettagli interni di Java, ma ha preferito seguire la strada del mondo UNIX mettendo a disposizione di tutti il codice. Del resto, il valore di Java non è dato dal codice, ma dall'idea di base.

Il codice sorgente dell'API è installato automaticamente quando si decomprime il JDK, ma in alcune piattaforme rimane in forma compressa in un file denominato `src.zip`, che si trova nella directory `java` creata al momento dell'installazione.

Figura 1.6
L'applet TicTacToe.

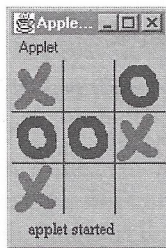
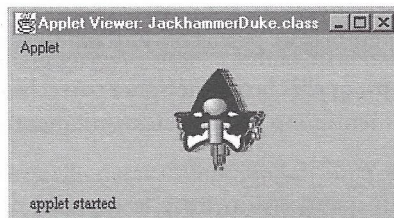



Figura 1.7
L'applet UnderConstruction.

Scelta di un IDE

Anche se il JDK è sufficiente per un'attività di sviluppo in Java a livello professionale, le funzioni avanzate degli IDE (Integrated Development Environment, ambienti di sviluppo integrati) possono migliorare notevolmente la produttività. In ogni caso, non è detto che tutti necessitino di un IDE; nella scelta del prodotto occorre considerare alcuni aspetti specifici relativi alle peculiarità di Java (come la portabilità).

- ✔ Qual è l'importanza della progettazione dell'interfaccia grafica per i propri programmi?
- ✔ I programmi devono essere del tutto portabili su varie piattaforme?
- ✔ Qual è il proprio livello di famigliarità con gli IDE?
- ✔ Si desidera poter programmare in altri linguaggi con lo stesso strumento?

Strumenti di sviluppo di GUI

I sistemi a finestre si sono ormai affermati definitivamente e gli utenti rifiutano il software che non presenta le corrispondenti funzionalità, anche se rimangono i nostalgici affezionati alle utilità del DOS, con il prompt dei comandi e i file batch. Oggi servono il controllo del mouse e le finestre ridimensionabili; poiché questi requisiti fanno della progettazione dell'interfaccia grafica un aspetto importante della maggior parte dei programmi Java, è opportuno scegliere un IDE ben corredato in questo campo.

La maggior parte degli IDE si distingue per l'approccio alla progettazione dell'interfaccia e alle funzionalità offerte al riguardo. Inoltre, il software come Rogue Wave JFactory è più uno strumento per lo sviluppo dell'interfaccia che un IDE completo.

Molti strumenti per la creazione dell'interfaccia funzionano più o meno come un programma di disegno: si inizia con un modulo vuoto e una serie di componenti dell'interfaccia; tali componenti sono solitamente oggetti dell'AWT (Abstract Windowing Toolkit), e il programma genera codice AWT che può essere modificato da chi è esperto nell'uso di questa funzionalità. Lo scopo è quello di facilitare la gestione delle finestre e dei componenti dell'interfaccia, in modo che il programmatore possa concentrarsi su aspetti più importanti.

Per alcuni programmatori, soprattutto quelli che hanno speso parecchio tempo nell'apprendere i dettagli dell'AWT, questi strumenti sono poco interessanti, mentre per altri si tratta di accesori utilissimi.

Una delle migliori caratteristiche di questi strumenti è la capacità di generare codice di gestione di eventi contemporaneamente alla creazione di un componente dell'interfaccia. Chi ha utilizzato Visual Basic dovrebbe sapere di che cosa si tratta: si posiziona un campo di testo dove desiderato, si fa doppio clic e poi si inserisce il codice sorgente che ne controlla il funzionamento.

Portabilità del codice

Una delle caratteristiche di cui tenere conto per la scelta di un IDE è il fatto che esso generi codice pienamente compatibile con la libreria di classi standard di Java; diversi ambienti di sviluppo come Café e Java WorkShop sono forniti con una versione particolare di Java, invece di basarsi su un'implementazione esistente. Solitamente non vi sono problemi, perché uno degli scopi degli IDE è quello di sfruttare la portabilità su qualsiasi piattaforma. Molti ambienti di sviluppo come SourceCraft NetCraft si basano interamente sul JDK per non utilizzare estensioni proprietarie.

Altri ambienti, come Microsoft Visual J++, forniscono estensioni specifiche per il sistema operativo Windows. In questo caso, per creare un applet o applicazione pienamente portabile occorre evitare l'utilizzo di queste estensioni. Poiché questo IDE è relativamente nuovo, occorre ancora verificare se i vantaggi apportati da queste estensioni compensano lo svantaggio di rimanere legati a una piattaforma specifica. Il problema è al centro di un acceso dibattito nella comunità degli sviluppatori Java, perché molti ritengono che il continuo aumento di popolarità del linguaggio dipenda dal fatto che esso rimanga unito e del tutto indipendente dalla piattaforma.

Un punto interessante legato alla portabilità è che la maggior parte degli IDE per Java non sono essi stessi indipendenti dalla piattaforma, anche quando si tratta di programmi Java. Tutti sono offerti per una piattaforma specifica, principalmente perché utilizzano codice nativo.

Esperienza nell'uso degli IDE

Un elemento degli IDE che talvolta provoca confusione è il livello di abilità richiesto per utilizzarli. Se lo scopo è quello di migliorare la programmazione, questo non può accadere se non si riesce a utilizzare l'ambiente. Un ambiente di sviluppo integrato è un software complesso, spesso basato su un'interfaccia a documenti multipli, dove si possono avere diverse finestre aperte contemporaneamente e numerosissime opzioni.

Per i programmatori esperti, si tratta di cose utili, poiché serve tutto il controllo possibile, mentre i principianti possono perdersi in tutte le opzioni disponibili. Chi è impegnato ad apprendere l'uso di un linguaggio, non ha tempo di pensare anche all'apprendimento dell'IDE. Molti IDE per Java sono ideali per gli esperti, mentre pochi sono adatti ai principianti grazie all'interfaccia meno complessa.

Il migliore esempio è quello di Java WorkShop di SunSoft, che utilizza la familiare interfaccia di un browser Web. Chi arriva a Java con un'esperienza di programmazione limitata, come gli sviluppatori in HTML che desiderano migliorare le proprie capacità, potrebbe trovare questi IDE più adatti alle proprie esigenze.

Naturalmente anche la facilità d'uso ha il rovescio della medaglia. Un ambiente come Java WorkShop potrebbe richiedere un procedimento più lungo per ottenere il risultato richiesto, perché le funzioni non sono immediatamente disponibili o perché l'ambiente potrebbe non offrire le funzionalità di un IDE più complesso.

Sviluppo in vari linguaggi

Un altro fattore importante nell'uso di un IDE per alcuni programmatori è la possibilità di impiegare altri linguaggi. Diversi ambienti di sviluppo, come Metrowerks CodeWarrior e Microsoft Visual J++, sono progettati per gestire vari linguaggi, o sono del tutto equivalenti agli altri strumenti di sviluppo della casa produttrice.

Nel caso di un IDE complesso (come questi), questa caratteristica è utile perché non serve apprendere l'uso di un altro ambiente quando si passa a un linguaggio diverso da Java. Inoltre, se si scrivono metodi nativi da utilizzare nei programmi Java, si possono utilizzare IDE multilinguaggio. L'ambiente Visual J++ è pienamente integrato con l'ambiente Visual C++, perciò è possibile scrivere metodi nativi in modo integrato.

Symantec Café

Questo programma, presentato per la prima volta nel marzo del 1996, è il primo ambiente di sviluppo reso disponibile per Java dopo il JDK. La Symantec parla di IDDE (Integrated Development and Debugging Environment), ma in realtà la "D" in più non fa molta differenza rispetto agli altri IDE, che anch'essi in genere includono un debugger. Café è basato sull'ambiente del C++ della Symantec, ma è un prodotto indipendente che non richiede la piattaforma del C++.

Requisiti di sistema

Esistono versioni di Café per Microsoft Windows 95, Windows NT 3.5x e Macintosh.

Per i sistemi Microsoft serve almeno un processore 386 con 8 M di memoria, ma si consiglia almeno un Pentium con 16 M. Serve un monitor VGA (consigliato SVGA), 60 M di spazio su disco e un lettore di CD-ROM.

Per i sistemi Macintosh serve un PowerMacintosh o un processore 68030 o superiore, 16 M di memoria e 30 M di spazio su disco.

Café comprende il JDK con tutte le librerie di classi e gli esempi di codice. In effetti, conviene rimuovere un'eventuale versione esistente del JDK prima di installare Café, per evitare conflitti.

Panoramica

Café è un IDE sofisticato che offre un eccellente editor di codice in cui la sintassi è evidenziata con il colore, un editor per modificare la gerarchia di classe, uno strumento per la progettazione dell'interfaccia e numerosi applet di esempio. Per facilitare il progetto della gerarchia di classi, Café dispone di un editor apposito che consente di navigare tra le classi e modificare i metodi di classe, oltre a un editor per visualizzare e modificare le relazioni tra le classi.

Le modifiche al codice sorgente che influiscono sulla gerarchia di classe sono visibili mentre il programma viene scritto, non occorre compilare. Si può anche modificare il codice sorgente direttamente dall'editor di classe: facendo clic su una funzione o su un metodo si apre una finestra con il codice sorgente.

Con AppExpress, la creazione della struttura del programma diventa più rapida. Questa e altre autocomposizioni (chiamate Express Agent in Café) facilitano l'inizio di nuovi progetti e programmi.

Con Café Studio, la progettazione dell'interfaccia utente grafica può essere svolta in modo visuale, con il mouse. Lo strumento crea automaticamente i gestori di eventi per i componenti. C'è anche un editor di menu con una finestra attiva in cui è possibile effettuare un collaudo. Le risorse sono salvate in file `.rc` separati che possono essere modificati in seguito (esattamente come i file sorgente). Il formato `.rc` è compatibile con altri strumenti di progettazione di questo tipo.

Un aspetto interessante di Café Studio è il fatto che offre la possibilità di progettare un modulo e determinarne con precisione l'aspetto. Con il JDK e il suo AWT, i progettisti di interfaccia devono consentire che il loro lavoro sia modificato in base alla piattaforma sulla quale si esegue l'applet o l'applicazione. È lo stesso approccio adottato con HTML, per cui il codice può essere modificato per adattarsi alle diverse piattaforme in uso nel World Wide Web. Con Café Studio, i programmatori possono scegliere tra i gestori di layout per indicare la posizione e la dimensione di tutti gli elementi dell'interfaccia.

Quando si è pronti per compilare il programma, Café consente di scegliere tra il compilatore della Sun e quello integrato, più veloce rispetto a quello standard del JDK.

Il debugger di Café mette a disposizione vari modi per sospendere temporaneamente l'esecuzione del codice, tra cui una funzione di breakpoint rapido per interrompere l'esecuzione a una riga specifica. Il debugger offre inoltre un ampio controllo sui thread in programmi multithreading. Una finestra consente di controllare il contenuto delle variabili.

L'ambiente di Symantec Café consente una notevole personalizzazione: barre e caselle degli strumenti possono essere ridimensionati e posizionati dove si desidera sullo schermo. È possibile aprire diverse finestre nello stesso tempo, ad esempio per osservare la gerarchia di oggetti mentre si inserisce il codice sorgente e si utilizza l'editor di moduli.

Come procurarselo

La home page di Symantec Café si trova al seguente URL:

```
http://cafe.symantec.com/
```

SunSoft Java WorkShop

Questo ambiente di sviluppo è scritto quasi interamente in Java, anche perché è stato sviluppato per migliorare il linguaggio stesso.

Requisiti di sistema

Esistono versioni di Java WorkShop per Microsoft Windows 95, Windows NT 3.5.1, SPARC Solaris (versione 2.4 e successive) e sistemi Solaris basati su Intel x86.

Nei sistemi Windows 95 e NT serve un processore Pentium a 90 MHz o superiore, con 16 M di memoria e 45 M di spazio su disco. Nei sistemi Solaris sono richiesti 32 M di memoria, 45 M di spazio su disco e un sistema a finestre compatibile OSF/Motif 1.2.3. La risoluzione consigliata per lo schermo è di 800X600 pixel.

Java WorkShop è fornito con una versione proprietaria del JDK, perciò non può essere utilizzato con il kit standard. Come per Café, occorre disinstallare un'eventuale copia del JDK prima di utilizzare Java WorkShop.

Panoramica

Java WorkShop è uno dei migliori ambienti di sviluppo per i principianti; utilizza un'interfaccia basata sul Web e offre un editor di codice sorgente, un browser di classi, un debugger, un gestore di progetti e Visual Java (uno strumento per il disegno visuale di un'interfaccia grafica).

La principale differenza tra Java WorkShop e altri ambienti è data dall'interfaccia, che in questo caso è più simile a quella di un browser Web che a quella di un IDE. In effetti si tratta proprio di un browser Web, come possono rilevare gli utenti di HotJava della Sun. Inoltre, con Java WorkShop si può visualizzare qualsiasi pagina Web.

L'interfaccia di Java WorkShop è comoda soprattutto per chi non ha familiarità con gli IDE, mentre per gli esperti può risultare un po' frustrante.

Il browser crea pagine HTML nello stesso formato della documentazione generata da `javadoc`. L'editor di codice lavora insieme al debugger, infatti gli errori di compilazione creano un collegamento che porta alla riga di codice interessata.

Visual Java consente di disegnare l'interfaccia grafica, come Café Studio; il programmatore può sviluppare finestre di dialogo e altri elementi visuali, e i relativi gestori di eventi sono creati automaticamente. C'è anche un editor di menu. I file di risorsa sono salvati in file `.gui` separati che possono essere modificati in seguito, esattamente come i file sorgente.

L'ambiente non è personalizzabile come Café, ma l'interfaccia facilita l'integrazione con altri strumenti. In pratica si tratta di una raccolta di pagine Web contenenti programmi Java.

Come procurarselo

Java WorkShop può essere prelevato gratuitamente a scopo di valutazione presso il seguente sito Web:

```
http://www.sun.com/sunsoft/Developer-products/java/index.html
```

Microsoft Visual J++

Visual J++ è l'ambiente di sviluppo per Java della Microsoft. Progettato in modo da integrarsi con la serie di strumenti di sviluppo Visual Studio, Visual J++ presenta estensioni alla libreria di classi di Java specifiche per la piattaforma Windows. L'ambiente è pressoché identico a quello del Visual C++ e fornisce il pieno supporto per l'integrazione di Java con ActiveX.

Requisiti di sistema

Attualmente Visual J++ è disponibile soltanto per la piattaforma Windows 95/NT; richiede almeno un processore 486 con 8 M di memoria per Windows 95 (12 M consigliati) e 16 M per Windows NT (20 M consigliati). L'installazione tipica richiede 20 M di spazio su disco, quella minima 14 M.

Panoramica

Per gli sviluppatori in ambiente Windows che hanno familiarità con Visual C++, il passaggio a Visual J++ è molto agevole. Anche per gli altri, l'ambiente non è particolarmente complesso. Visual J++ comprende delle autocomposizioni che guidano passo passo nella creazione di applet.

L'editor è di ottimo livello ed evidenzia la sintassi con i colori. C'è anche un visualizzatore di classi, proprietà e metodi. Il debugger grafico consente di controllare diversi applet contemporaneamente da un browser e comprende un disassemblatore di bytecode, la possibilità di tracciare le istruzioni e di assegnare valori alle variabili durante il debugging.

Tutto considerato, si tratta di uno dei migliori IDE disponibili; alcuni sono preoccupati per le estensioni proprietarie a Java, ma la Microsoft sa di dover aderire alle funzioni fondamentali dell'API Java, perciò non vi sono grossi problemi.

Come procurarselo

Per informazioni su come ottenere Microsoft Visual J++ occorre collegarsi alla home page della Microsoft:

```
http://www.microsoft.com/visualj/
```

Asymetrix SuperCede

SuperCede è un IDE basato sui moduli, creato allo scopo di facilitare chi si avvicina a Java per la prima volta. La stretta integrazione tra moduli ed editor di codice facilita la navigazione tra gli oggetti dell'interfaccia utente e i relativi gestori di eventi. Il sistema comprende un'interessante tecnologia di compilazione *flash*, che consente allo sviluppatore di modificare il codice sorgente e osservare immediatamente l'impatto della modifica. Inoltre è possibile scegliere se generare bytecode portabile o codice nativo per macchine Intel, che offre prestazioni migliori.

Requisiti di sistema

SuperCede è attualmente disponibile soltanto per la piattaforma Windows 95/NT; richiede almeno un 486/66 con 32 M di memoria e 50 M di spazio su disco.

Panoramica

Questo ambiente di sviluppo integrato comprende un gestore di progetti, un editor che evidenzia a colori la sintassi e un editor di moduli basato sul mouse, oltre a un debugger grafico. Inoltre vi sono varie finestre per lo scorrimento di codice sorgente, moduli e file di classe.

Una peculiarità del sistema è il compilatore flash, che consente di modificare interattivamente le applicazioni in esecuzione. In più, questo compilatore consente di scegliere se generare bytecode Java standard o eseguibili nativi x86 che offrono prestazioni migliori, ma sono legati alla piattaforma Intel.

Roaster

Roaster della Roaster Technology è stato presentato nel gennaio del 1996 come il primo IDE Java per Macintosh. La versione 2.3 presenta notevoli miglioramenti rispetto alla prima; comprende uno strumento per la creazione dell'interfaccia con il mouse, un compilatore che può essere indirizzato a sistemi Macintosh o Windows e un'estesa libreria di classi.

Requisiti di sistema

I requisiti minimi sono un processore 68030 o superiore, Macintosh System 7.1.2 o successivo (consigliato 7.5 o successivo) e 8 M di RAM.

Panoramica

Roaster rappresenta la scelta migliore per gli utenti Macintosh, principalmente perché è stato il primo IDE per Macintosh e quindi ha avuto più tempo di evolversi. Comprende un browser integrato, un visualizzatore di classi e un editor. Supporta pienamente AppleScript e comprende un'autocomposizione che guida nella creazione di classi AppleScript.

Il sistema offre il supporto per eseguire e collaudare applet in rete, e classi Java per accedere a database con ODBC. È l'unico IDE che offre agli utenti registrati un abbonamento gratuito alla rivista Java Report, della SIGS.

Come procurarselo

Roaster si trova presso la home page della Roaster Technologies presso:

```
http://www.roaster.com/roaster/
```

SourceCraft NetCraft

La SourceCraft, la casa che ha prodotto l'ambiente di sviluppo ObjectCraft, offre l'IDE NetCraft a titolo gratuito. Si tratta di un programma *freeware*, ovvero disponibile gratuitamente per chi rispetta i termini d'uso.

Requisiti di sistema

Esistono versioni per Microsoft Windows 95 e NT 3.5.1, che richiedono un processore 486 o superiore con almeno 8 M di memoria. Il sistema è fornito con la versione corrente del JDK; è anche possibile prelevare il solo programma, senza il JDK, se si dispone già del kit.

Panoramica

SourceCraft NetCraft ha meno funzionalità di altri IDE, tuttavia crea programmi Java compatibili con tutte le implementazioni del linguaggio. Contiene un editor, un ispettore di classi, uno strumento per la progettazione dell'interfaccia utente e un compilatore.

Si tratta di un IDE utilizzabile per qualunque tipo di applet o applicazioni, pensato soprattutto per l'uso in ambienti intranet.

Come procurarselo

Per ulteriori informazioni si consiglia di collegarsi alla home page di NetCraft presso:

```
http://www.sourcecraft.com:4800/about/netcraft/
```

Pro-C WinGEN per Java

WinGEN per Java è un IDE pensato per chi non è un programmatore. Lo scopo è quello di creare automaticamente il codice in modo che persino chi non ha mai programmato sia in grado di ottenere applicazioni e applet Java. L'interfaccia utente grafica può essere creata con il mouse, ed è possibile introdurre animazioni senza scrivere nemmeno una riga di codice. La compilazione e l'esecuzione sono affidate al Java Development Kit.

Requisiti di sistema

WinGEN è disponibile per sistemi Windows 95 e NT con un processore 486 o superiore, 8 M di memoria e 10 M di spazio su disco. WinGEN Lite, la versione di valutazione richiede un programma in grado di decomprimere un archivio ZIP rispettando i nomi di file lunghi, altrimenti l'installazione fallisce. È compresa la versione corrente del JDK, che occorre installare prima di WinGEN.

Panoramica

WinGEN rappresenta un potenziamento del JDK, più che un sostituto, poiché consente di eseguire il compilatore e l'interprete del JDK invece di utilizzare la riga di comando.

Trattandosi di un software dedicato ai principianti, WinGEN punta molto sull'uso del mouse per la creazione dell'interfaccia utente. Molti eventi possono essere creati senza scrivere nemmeno una riga di codice.

L'interfaccia utente ricorda quella di altri strumenti per la creazione di interfacce, e risulta familiare soprattutto agli utenti di Visual Basic. Lo sviluppo è centrato sulle risorse: invece di iniziare dal codice e utilizzarlo per creare menu e finestre di dialogo, si inizia con questi ultimi e si genera poi il codice richiesto. Manca la possibilità di vedere l'interfaccia prima di generare il codice: il programma Java che utilizza l'interfaccia deve essere compilato ed eseguito prima che si possa osservare l'aspetto dell'interfaccia stessa.

La versione commerciale comprende alcune funzionalità non comuni in altri IDE, come il supporto per specifici database e tabelle di testo ASCII.

Un sistema denominato CodeHooks gestisce la programmazione avanzata, richiesta per scrivere codice allo scopo di gestire situazioni particolari che WinGEN non è in grado di gestire. I blocchi di codice che eseguono compiti specifici sono mantenuti separati dal codice generato automaticamente da WinGEN, perciò i programmatori possono modificare l'interfaccia senza dover riscrivere tutto il codice.

Come procurarselo

Per ulteriori informazioni su WinGEN e WinGEN Lite si consiglia di visitare la seguente pagina Web:

```
http://www.pro-c.com/products/wfj/java.html
```

Rogue Wave JFactory

A differenza degli altri programmi esaminati finora, JFactory è offerto come strumento per la creazione di interfacce piuttosto che come IDE. Tuttavia, poiché JFactory consente di inserire il codice di gestione di eventi dall'interno del programma e anche di compilare e collaudare i programmi durante lo sviluppo, è in pratica molto vicino a un IDE vero e proprio. Anche se JFactory può essere utilizzato insieme a qualsiasi editor e compilatore Java, è fornito un editor di default e inizialmente il programma è impostato per l'uso del compilatore del JDK.

Requisiti di sistema

Esistono versioni di JFactory per le seguenti piattaforme:

- ✔ Microsoft Windows 95 e Windows NT con processore 486 o superiore, 16 M di memoria, 25 M di spazio su disco e il JDK versione 1.02.

- ✔ SPARC Solaris 2.4 o 2.5 con UNIX, 25 M di spazio su disco, un browser per gli applet, JDK versione 1.02 e X11R5. Occorre avere memoria sufficiente per eseguire X11R5, il compilatore del JDK e un browser per gli applet.
- ✔ HP-UX 10.01 con 25 M di spazio su disco, un browser di applet, JDK versione 1.02 e X11R5.
- ✔ IBM OS/2 Warp 3.0 con 25 M di spazio su disco, 4 M di memoria che non sia utilizzata da OS/2, un mouse a due pulsanti o altro dispositivo di puntamento e il JDK versione 1.02 build os2-19960412.

Panoramica

JFactory è un sofisticato strumento per la creazione di interfacce basato su zApp Factory, una struttura di applicazione C++ multipiattaforma della Rogue Wave. Presenta numerose funzionalità che facilitano una rapida progettazione dell'applicazione, e trae beneficio dall'esperienza maturata dalla Rogue Wave con altri strumenti di programmazione. L'impiego principale è quello di creare interfacce in modo visuale, con il mouse, campo in cui è nettamente superiore agli altri concorrenti.

JFactory non è considerato un IDE vero e proprio perché non offre un compilatore, un debugger o altri strumenti, tuttavia l'editor visuale è talmente facile da utilizzare e così potente che può compensare la mancanza di altre funzionalità. Si tratta di un ambiente aperto, che consente di utilizzare qualsiasi compilatore o debugger.

Come procurarselo

Per ulteriori informazioni su JFactory e per prelevare una versione dimostrativa occorre collegarsi al sito:

```
http://www.roguewave.com/products/jfactory/jfactory.html
```

Metrowerks CodeWarrior

Questo è l'unito IDE che supporta altri tre linguaggi di programmazione (C, C++ e Pascal) nello stesso ambiente. Inoltre, questi linguaggi sono supportati sia su Macintosh sia su Windows 95/NT. In questo modo gli sviluppatori Java possono utilizzare altri linguaggi senza acquistare altro software.

Requisiti di sistema

I requisiti minimi per la versione Macintosh di CodeWarrior sono un processore 68020 o superiore (oppure PowerPC 601 o superiore), System 7.1 o successivo, 8 M di memoria RAM (consigliati 16 M) e 65 M di spazio su disco. La versione per Windows richiede un processore 486 o superiore, Windows 95/NT e 16 M di RAM.

Panoramica

CodeWarrior si è affermato nel mondo Macintosh come ambiente di sviluppo per C/C++, perciò non sorprende il successo che sta riscuotendo anche con Java, almeno sul Mac. L'IDE comprende un gestore di progetti, un editor di risorse, un editor di testo, un debugger grafico e vari sistemi per la visualizzazione delle classi.

L'ampio supporto offerto agli sviluppatori C/C++ che passano a Java costituisce senz'altro un punto di forza, almeno in ambiente Macintosh, mentre per la piattaforma Windows la situazione è diversa.

Come procurarselo

Ulteriori informazioni su CodeWarrior sono disponibili presso il sito della Metrowerks:

```
http://www.metrowerks.com/products/
```

Riepilogo

Il Java Development Kit fornisce gli strumenti essenziali per programmare in Java e numerosa documentazione. Questi strumenti saranno esaminati meglio nel seguito del libro, ma è importante comprenderne il ruolo fin da subito.

In questo capitolo sono inoltre stati esaminati altri ambienti di sviluppo integrati che possono facilitare notevolmente lo sviluppo in Java, affinché il lettore possa scegliere quello più adatto alle proprie esigenze.